(72) Inventor; and
(75) Inventor/Applicant (for US only): JARMAN, Nick [GB/GB]; 7 Cadet Way, Church Crookham, Fleet, Hampshire GU52 8UG (GB).

(54) Title: IMAGE PROCESSING TO REMOVE RED-EYE FEATURES
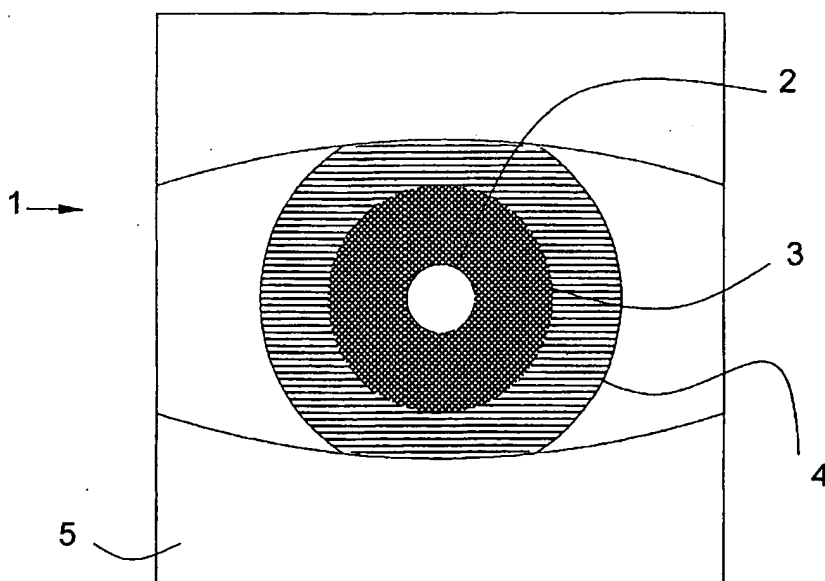
(57) Abstract: A method of processing a digital image to detect and remove red-eye features includes identifying highlight regions of the image having pixels with higher saturation and/or lightness values than pixels in the regions therearound, associating red-eye features with at least some of the highlight regions, and performing red-eye reduction on at least some of said red-eye features. Further selection criteria may be applied to red-eye features before red-eye reduction is carried out.

IMAGE PROCESSING TO REMOVE RED-EYE FEATURES

This invention relates to the detection and reduction of red-eye in digital images.

5   The phenomenon of red-eye in photographs is well-known. When a flash is used to illuminate a person (or animal), the light is often reflected directly from the subject's retina back into the camera. This causes the subject's eyes to appear red when the photograph is displayed or printed.

10   Photographs are increasingly stored as digital images, typically as arrays of pixels, where each pixel is normally represented by a 24-bit value. The colour of each pixel may be encoded within the 24-bit value as three 8-bit values representing the intensity of red, green and blue for that pixel. Alternatively, the array of pixels can be transformed so that the 24-bit value consists of three 8-bit values representing "hue",
15   "saturation" and "lightness". Hue provides a "circular" scale defining the colour, so that 0 represents red, with the colour passing through green and blue as the value increases, back to red at 255. Saturation provides a measure of the intensity of the colour identified by the hue. Lightness can be seen as a measure of the amount of illumination.
20

By manipulation of these digital images it is possible to reduce the effects of red-eye. Software which performs this task is well known, and generally works by altering the pixels of a red-eye feature so that their red content is reduced – in other words so that their hue is rendered less red. Normally they are left as black or dark grey instead.
25

Most red-eye reduction software requires the centre and radius of each red-eye feature which is to be manipulated, and the simplest way to provide this information is for a user to select the central pixel of each red-eye feature and indicate the radius of the red part. This process can be performed for each red-eye feature, and the manipulation
30   therefore has no effect on the rest of the image. However, this requires considerable input from the user, and it is difficult to pinpoint the precise centre of each red-eye feature, and to select the correct radius. Another common method is for the user to

draw a box around the red area. This is rectangular, making it even more difficult to accurately mark the feature.

There is therefore a need to identify automatically areas of a digital image to which red-
5   eye reduction should be applied, so that red-eye reduction can be applied only where it is needed, either without the intervention of the user or with minimal user intervention.

The present invention recognises that a typical red-eye feature is not simply a region of red pixels. A typical red-eye feature usually also includes a bright spot caused by
10   reflection of the flashlight from the front of the eye. These bright spots are known as "highlights". If highlights in the image can be located then red-eyes are much easier to identify automatically. Highlights are usually located near the centre of red-eye features, although sometimes they lie off-centre, and occasionally at the edge.

15   In accordance with a first aspect of the present invention there is provided a method of processing a digital image, the method comprising:
        identifying highlight regions of the image having pixels with higher saturation and/or lightness values than pixels in the regions therearound;
        identifying red-eye features associated with some or all of said highlight regions;
20   and
        performing red-eye reduction on some or all of the red-eye features.

This has the advantage that the saturation/lightness contrast between highlight regions and the area surrounding them is much more marked than the colour (or "hue") contrast
25   between the red part of a red-eye feature and the skin tones surrounding it. Furthermore, colour is encoded at a low resolution for many image compression formats such as JPEG. By using saturation and lightness to detect red-eyes it is much less likely that they will be missed than if hue is used as the basic detection tool.

30   It is convenient if each red-eye feature can have a unique reference point associated with it, to enable the location of the red-eye feature to be stored in a list. A single reference pixel in each highlight region may therefore be selected as the central point

3

for the red-eye feature associated with that highlight region, and the red-eye reduction for that red-eye feature centred on the reference pixel.

As well as having high saturation and/or lightness values, the highlight of a typical red-
5   eye feature is very sharply defined. Accordingly a highlight region is preferably only identified if there is a sharp change in pixel saturation and/or lightness between the highlight region and the regions adjacent thereto.

Although many of the identified highlight regions may result from red-eye, it is likely
10   that some highlight regions will be identified which are not part of red-eye features, and around which a red-eye reduction should not be applied. The method therefore preferably comprises eliminating at least some of the highlight regions as possibilities for red-eye reduction. Indeed, it is possible that none of the highlight regions identified are caused by red-eye, and therefore should not have red-eye features associated with
15   them. In this context it will be appreciated that the phrase "identifying red-eye features with some or all of said highlight regions" is intended to include the possibility that no red-eye features are associated with any of the highlight regions. Similarly, it is possible that filters applied to red-eye features determine that none of the red-eye features originally identified should have red-eye reduction applied to them, and
20   accordingly the phrase "performing red-eye reduction on some or all of the red-eye features" includes the possibility that all red-eye features are rejected as possibilities for red-eye reduction

In practice, there is a maximum size that a red-eye feature can be, assuming that at least
25   an entire face has been photographed. Therefore, preferably, if a highlight region exceeds a predetermined maximum diameter no red-eye feature is associated with that highlight region, and no red-eye reduction is carried out.

Red-eye features are generally substantially circular. Therefore linear highlight features
30   will in general not be due to red-eye, and therefore preferably no red-eye reduction is performed on a feature associated with a highlight region if that highlight region is substantially linear.

4

Red-eye reduction is preferably not carried out on any red-eye features which overlap each other.

Once the highlight regions have been determined, it is convenient to identify the hue of pixels in the region surrounding each highlight region, and only perform red-eye reduction for a red-eye feature associated with a highlight region if the hue of the pixels surrounding that highlight region contains more than a predetermined proportion of red. The radius of the red-eye feature can then be determined from this region of red pixels surrounding the highlight region. Red-eye reduction is preferably only performed on a red-eye feature if the ratio of radius of the red-eye region to the radius of the highlight region falls within a predetermined range of values. For a typical red-eye feature, the radius of the red-eye region will be up to 8 times the radius of the highlight region.

Preferably, assuming that the digital image is derived from a photograph, it is determined whether a flash was fired when the photograph was taken, and highlight regions are not identified or red-eye reduction performed if no flash was fired.

It is preferably determined whether the digital image is monochrome, and, if so, highlight regions are not identified or red-eye reduction performed.

In some cases, for example in portrait photography, the user may know in advance that all highlights will be caused by red-eye, in which case a red-eye feature may be associated with each highlight region identified, and red-eye reduction may be carried out on all red-eye features.

In accordance with a second aspect of the present invention there is provided a method of detecting red-eye features in a digital image, comprising:

identifying highlight regions comprising pixels having higher saturation and/or lightness values than pixels in the regions therelaround; and

determining whether each highlight region corresponds to a red-eye feature on the basis of applying further selection criteria.

The further selection criteria preferably include testing the hue of pixels surrounding the highlight region, and determining that the highlight region does not correspond to a red-eye feature if said hue is outside a predetermined range corresponding to red.

5   The further selection criteria may alternatively or in addition include identifying the shape of the highlight region, and determining that the highlight region does not correspond to a red-eye feature if said shape is not substantially circular.

In accordance with a third aspect of the invention there is provided a method of
10   reducing the visual effect of red-eye features in a digital image, comprising detecting red-eye features using the method described above, and changing the hue of pixels around each highlight region to reduce the red content of those pixels.

The invention also provides a digital image to which the method described above has
15   been applied, apparatus arranged to perform the method, and a computer storage medium having stored thereon a computer program arranged to perform the method.

Some preferred embodiments of the invention will now be described by way of example only and with reference to the accompanying drawings, in which:

20
Figure 1 is a flowchart describing a general procedure for reducing red-eye;

Figure 2 is a schematic diagram showing a typical red-eye feature;

25   Figure 3 shows the red-eye feature of Figure 2, showing pixels identified in the detection of a highlight;

Figure 4 shows the red-eye feature of Figure 2 after measurement of the radius; and

30   Figure 5 is a flowchart describing a procedure for detecting red-eye features.

When processing a digital image which may or may not contain red-eye features, in order to correct for such features as efficiently as possible, it is useful to apply a filter to

6

determine whether such features could be present, find the features, and apply a red-eye correction to those features, preferably without the intervention of the user.

In its very simplest form, an automatic red-eye filter can operate in a very
5   straightforward way. Since red-eye features can only occur in photographs in which a flash was used, no red-eye reduction need be applied if no flash was fired. However, if a flash was used, or if there is any doubt as to whether a flash was used, then the image should be searched for features resembling red-eye. If any red-eye features are found, they are corrected. This process is shown in Figure 1.
10

An algorithm putting into practice the process of Figure 1 begins with a quick test to determine whether the image could contain red-eye: was the flash fired? If this question can be answered 'No' with 100% certainty, the algorithm can terminate; if the flash was not fired, the image cannot contain red-eye. Simply knowing that the flash did not fire
15   allows a large proportion of images to be filtered with very little processing effort.

There are a number of possible ways of determining whether the flash was fired. One method involves asking the user, although this is not ideal because it involves user interaction, and the user may not be able to answer the question reliably.
20

Another alternative involves looking in the image metadata. For example, an EXIF format JPEG has a 'flash fired – yes/no' field. This provides a certain way of determining whether the flash was fired, but not all images have the correct metadata. Metadata is usually lost when an image is edited. Scanned images containing red-eye
25   will not have appropriate metadata.

There is an additional method of determining if the flash was fired, which is appropriate if the algorithm is implemented in the controlling software of a digital camera. The module responsible for taking the picture could indicate to the red-eye
30   detection/correction module that the flash was fired.

7

For any image where it cannot be determined for certain that the flash was not fired, a more detailed examination must be performed using the red-eye detection module described below.

5   If no red-eye features are detected, the algorithm can end without needing to modify the image. However, if red-eye features are found, each must be corrected using the red-eye correction module described below.

Once the red-eye correction module has processed each red-eye feature, the algorithm

10   ends.

The output from the algorithm is an image where all detected occurrences of red-eye have been corrected. If the image contains no red-eye, the output is an image which looks substantially the same as the input image. It may be that the algorithm detected

15   and 'corrected' features on the image which resemble red-eye closely, but it is quite possible that the user will not notice these erroneous 'corrections'.

The red-eye detection module will now be described.

20   Figure 2 is a schematic diagram showing a typical red-eye feature 1. At the centre of the feature 1 is a white or nearly white "highlight" 2, which is surrounded by a region 3 corresponding to the subject's pupil. In the absence of red-eye, this region 3 would normally be black, but in a red-eye feature this region 3 takes on a reddish hue. This can range from a dull glow to a bright red. Surrounding the pupil region 3 is the iris 4,

25   some or all of which may appear to take on some of the red glow from the pupil region 3.

The detection algorithm must locate the centre of each red-eye feature and the extent of the red area around it.

30

The red-eye detection algorithm begins by searching for regions in the image which could correspond to highlights 2 of red-eye features. The image is first transformed so that the pixels are represented by hue, saturation and lightness values. Most of the

8

pixels in the highlight 2 of a red-eye feature 1 have a very high saturation, and it is unusual to find areas this saturated elsewhere on facial pictures. Similarly, most red-eye highlights 2 will have high lightness values. It is also important to note that not only will the saturation and lightness values be high, but also they will be significantly higher than the regions 3, 4, 5 immediately surrounding them. The change in saturation from the red pupil region 3 to the highlight region 2 is very abrupt.

The highlight detection algorithm scans each row of pixels in the image, looking for small areas of light, highly saturated pixels. During the scan, each pixel is compared with its preceding neighbour (the pixel to its left). The algorithm searches for an abrupt increase in saturation and lightness, marking the start of a highlight, as it scans from the beginning of the row. This is known as a "rising edge". Once a rising edge has been identified, that pixel and the following pixels (assuming they have a similarly high saturation and lightness) are recorded, until an abrupt drop in saturation is reached, marking the other edge of the highlight. This is known as a "falling edge". After a falling edge, the algorithm returns to searching for a rising edge marking the start of the next highlight.

A typical algorithm might be arranged so that a rising edge is detected if:
1. The pixel is highly saturated (saturation > 128).
2. The pixel is significantly more saturated than the previous one (this pixel's saturation – previous pixel's saturation > 64).
3. The pixel has a high lightness value (lightness > 128).

The rising edge is located on the pixel being examined. A falling edge is detected if:
1. The pixel is significantly less saturated than the previous one (previous pixel's saturation – this pixel's saturation > 64).
2. The previous pixel has a high lightness value (lightness > 128).

The falling edge is located on the pixel preceding the one being examined.

An additional check is performed while searching for the falling edge. After a defined number of pixels (for example 10) have been examined without finding a falling edge,

the algorithm gives up looking for the falling edge. The assumption is that there is a maximum size that a highlight in a red-eye feature can be – obviously this will vary depending on the size of the picture and the nature of its contents (for example, highlights will be smaller in group photos than individual portraits at the same

5   resolution). The algorithm may determine the maximum highlight width dynamically, based on the size of the picture and the proportion of that size which is likely to be taken up by a highlight (typically between 0.25% and 1% of the picture's largest dimension).

10   If a highlight is successfully detected, the co-ordinates of the rising edge, falling edge and the central pixel are recorded.

The algorithm is as follows:

```
for each row in the bitmap
        looking for rising edge = true
        loop from 2nd pixel to last pixel
                if looking for rising edge
                        if saturation of this pixel > 128 and..
                        ..this pixel's saturation - previous pixel's saturation > 64 and..
                        ..lightness of this pixel > 128 then
                                rising edge = this pixel
                                looking for rising edge = false
                        end if
                else
                        if previous pixel's saturation-this pixel's saturation > 64 and..
                        ..lightness of previous pixel > 128 then
                                record position of rising edge
                                record position of falling edge (previous pixel)
                                record position of centre pixel
                                looking for rising edge = true
                        end if
                end if

                if looking for rising edge = false and..
                ..rising edge was detected more than 10 pixels ago
                        looking for rising edge = true
                end if
        end loop
end for
```

The result of this algorithm on the red-eye feature 1 is shown in Figure 3. For this feature, since there is a single highlight 2, the algorithm will record one rising edge 6, one falling edge 7 and one centre pixel 8 for each row the highlight covers. The highlight 2 covers five rows, so five central pixels 8 are recorded. In Figure 3,

45   horizontal lines stretch from the pixel at the rising edge to the pixel at the falling edge. Circles show the location of the central pixels 8.

The location of all of these central pixels are recorded into a list of highlights which may potentially be caused by red-eye. The number of central pixels 8 in each highlight is then reduced to one. As shown in Figure 3, there is a central pixel 8 for each row

5   covered by the highlight 2. This effectively means that the highlight has been detected five times, and will therefore need more processing than is really necessary. It is therefore desirable to eliminate from the list all but the vertically central point from the list of highlights.

10   Not all of the highlights identified by the algorithm above will necessarily be formed by red-eye features. Others could be formed, for example, by light reflected from corners or edges of objects. The next stage of the process therefore attempts to eliminate such highlights, so that red-eye reduction is not performed on features which are not actually red-eye features.

15

There are a number of criteria which can be applied to recognise red-eye features as opposed to false features. One is to check for long strings of central pixels in narrow highlights – i.e. highlights which are essentially linear in shape. These may be formed by light reflecting off edges, for example, but will never be formed by red-eye.

20

This check for long strings of pixels may be combined with the reduction of central pixels to one. An algorithm which performs both these operations simultaneously may search through highlights identifying "strings" or "chains" of central pixels. If the aspect ratio, which is defined as the length of the string of central pixels 8 (see Figure 3)

25   divided by the largest width between the rising edge 6 and falling edge 7 of the highlight, is greater than a predetermined number, and the string is above a predetermined length, then all of the central pixels 8 are removed from the list of highlights. Otherwise only the central pixel of the string is retained in the list of highlights.

30

In other words, the algorithm performs two tasks:

•   removes roughly vertical chains of highlights from the list of highlights, where the aspect ratio of the chain is greater than a predefined value, and

o   removes all but the vertically central highlight from roughly vertical chains of highlights where the aspect ratio of the chain is less than or equal to a pre-defined value.

5   An algorithm which performs this combination of tasks is given below:

```
for each highlight
        (the first section deals with determining the extent of the chain of
        highlights - if any - starting at this one)

        make 'current highlight' and 'upper highlight' = this highlight
        make 'widest radius' = the radius of this highlight

        loop
                search the other highlights for one where: y co-ordinate =
                current highlight's y co-ordinate + 1; and x co-ordinate =
                current highlight's x co-ordinate (with a tolerance of ±1)

                if an appropriate match is found
                        make 'current highlight' = the match

                        if the radius of the match > 'widest radius'
                                make 'widest radius' = the radius of the match
                        end if
                end if
        until no match is found

        (at this point, 'current highlight' is the lower highlight in the chain
        beginning at 'upper highlight', so in this section, if the chain is
        linear, it will be removed; if it is roughly circular, all but the
        central highlight will be removed)

        make 'chain height' = current highlight's y co-ordinate - top
        highlight's y co-ordinate
        make 'chain aspect ratio' = 'chain height' / 'widest radius'

        if 'chain height' >= 'minimum chain height' and 'chain aspect ratio' >
        'minimum chain aspect ratio'
                remove all highlights in the chain from the list of highlights
        else
                if 'chain height' > 1
                        remove all but the vertically central highlight in the
                        chain from the list of highlights
                end if
        end if
end for
```

A suitable threshold for 'minimum chain height' is three and a suitable threshold for 'minimum chain aspect ratio' is also three, although it will be appreciated that these can be changed to suit the requirements of particular images.

Another criterion involves checking the hue of the pixels in the pupil region 3 around the highlight. If the pixels in this region contain less than a certain proportion of red then the feature cannot be red-eye. A suitable filter to apply to the pupil region 3 is that

12

unless the saturation is greater than or equal to 80 and the hue between 0 and 10, or between 220 and 255 (both inclusive) for 45% of the pixels around the highlight, then no red-eye reduction is performed on that feature.

5   The radius of the pupil region must then be established so that the extent of the red-eye feature is known, so that red-eye reduction can be performed. A suitable algorithm iterates through each highlight, roughly determining the radius of the red area which surrounds it. Once the algorithm has been completed, all highlights have an additional piece of information associated with them: the radius of the red-eye region. Therefore,

10   while the input to the algorithm is a series of highlights, the output can be considered to be a series of red-eye features.

The output may contain fewer red-eye regions than input highlights. In general, the ratio of the radius of the pupil region 2 to the radius of the highlight region 3 will

15   always fall within a certain range. If the ratio falls outside this range then it is unlikely that the feature being examined is due to red-eye. In the algorithm described, if the radius of the pupil region 3 is more than eight.times the radius of the highlight 2, the feature is judged not to be a red-eye feature, so it is removed from the list of areas to correct. This ratio has been determined by analysing a number of pictures, but it will be

20   appreciated that it may be possible to choose a different ratio to suit particular circumstances.

The method of determining the radius of the red area errs towards larger radii (because it only uses hue data, and does not take into account saturation or lightness) – in other

25   words, it calculates the area to be slightly larger than it actually is, meaning that it should contain all red pixels, plus some peripheral non-red ones, as shown in Figure 4. This is not a limitation as long as the method used for correcting the red-eye does not attempt to adjust non-red pixels. The slightly excessive size is also useful in the described embodiment, where no attempt is made to accurately determine the position

30   of the highlight within the red-eye region: the implementation of the embodiment assumes it is central, whereas this may not always be the case.

A suitable algorithm is given below:

13

```
for each highlight
        make 'calculated radius' = 0

        loop through the pixel rows in the image from this highlight's y co-
        ordinate - 'radius sample height' to this highlight's y co-ordinate +
        'radius sample height'
                scan the pixels leftwards and rightwards from the highlight to
                find the points at which the hue is outside the range of reds

                if half the distance between the two points > 'calculated radius'
                then
                        make 'calculated radius' half the distance between the two
                        points
                end if
        end loop

        if 'calculated radius' > 8 times the radius of the highlight
                remove this highlight from the list of highlights
        else
                record the calculated radius; the highlight is now a red-eye
                region
        end if
end for
```

It will be appreciated that this algorithm determines the radius of the red-eye feature by searching horizontally along rows of pixels centred on the highlight (which is defined as the central pixel 8 in a vertical row, as described above). The skilled person would be able to modify the algorithm to search radially from the highlight, or to determine the shape and extent of the red area surrounding the highlight.

Once the radii of red-eye features have been determined, a search can be made for overlapping features. If the red pupil region 3 overlaps with another red pupil region 3 around a highlight, then neither feature can be due to red-eye. Such features can therefore be discarded.

An algorithm to perform this task proceeds in two stages. The first iterates through all red-eye regions. For each red-eye region, a search is made until one other red-eye region is found which overlaps it. If an overlap is found, both red-eye regions are marked for deletion. It is not necessary to determine whether the red-eye region overlaps with more than one other.

The second stage deletes all red-eye regions which have been marked for deletion. Deletion must be separated from overlap detection because if red-eye regions were

deleted as soon as they were determined to overlap, it could clear overlaps with other red-eye regions which had not yet been detected.

The algorithm is as follows:

```
for each red-eye region
        search the other red-eye regions until one is found which overlaps this
        one, or all red-eye regions have been searched without finding an
        overlap

        if an overlap was found
                mark both red-eye regions for deletion
        end if
end for

loop through all red-eye regions
        if this region is marked for deletion
                delete it
        end if
end if
```

Two red-eye regions are judged to overlap if the sum of their radii is greater than the distance between their centres.

An alternative way of achieving the same effect as the algorithm above is to create a new list of red-eye features containing only those regions which do not overlap. The original list of red eye features can then be discarded and the new one used in its place.

The overall detection process is shown as a flow chart in Figure 5.

Red-eye reduction is then carried out on the detected red-eye features. There are a number of known methods for performing this, and a suitable process is now described. The process described is a very basic method of correcting red-eye, and the skilled person will recognise that there is scope for refinement to achieve better results, particularly with regard to softening the edges of the corrected area and more accurately determining the extent of the red-eye region.

There are two parts to the red-eye correction module: the controlling loop and the red-eye corrector itself. The controlling loop simply iterates through the list of red-eye regions generated by the red-eye detection module, passing each one to the red-eye corrector:

```
for each red-eye region
```

```
        correct red-eye in this region
end for
```

The algorithm for the red-eye corrector is as follows:

```
for each pixel within the circle enclosing the red-eye region
        if the saturation of this pixel >= 80 and...
        ...the hue of this pixel >= 220 or <= 10 then
                set the saturation of this pixel to 0

                if the lightness of this pixel < 200 then
                        set the lightness of this pixel to 0
                end if
        end if
end for
```

For each pixel, there are two very straightforward checks, each with a straightforward action taken as a consequence:

1. If the pixel is of medium or high saturation , and if the hue of the pixel is within the range of reds, the pixel is de-saturated entirely. In other words, saturation is set to "0" which causes red pixels to become grey.

2. Furthermore, if the pixel is dark or of medium lightness, turn it black. In most cases, this actually cancels out the adjustment made as a result of the first check: most pixels in the red-eye region will be turned black. Those pixels which are not turned black are the ones in and around the highlight. These will have had any redness removed from them, so the result is an eye with a dark black pupil and a bright white highlight.

A feature of the correction method is that its effects are not cumulative: after correction is applied to an area, subsequent corrections to the same area will have no effect. This would be a desirable feature if the red-eye detection module yielded a list of potentially overlapping red-eye regions (for example, if the multiple highlight detections were not eliminated). However, because overlapping red-eye regions are specifically removed, the non-cumulative nature of the correction module is not important to the current implementation.

It will be appreciated that the detection module and correction module can be implemented separately. For example, the detection module could be placed in a digital camera or similar, and detect red-eye features and provide a list of the location of these

features when a photograph is taken. The correction module could then be applied after the picture is downloaded from the camera to a computer.

The method according to the invention provides a number of advantages. It works on a whole image, although it will be appreciated that a user could select part of an image to which red-eye reduction is to be applied, for example just a region containing faces. This would cut down on the processing required. If a whole image is processed, no user input is required. Furthermore, the method does not need to be perfectly accurate. If red-eye reduction is performed around a highlight not caused by red-eye, it is unlikely that a user would notice the difference.

Since the red-eye detection algorithm searches for light, highly saturated points before searching for areas of red, the method works particularly well with JPEG-compressed images and other formats where colour is encoded at a low resolution.

It will be appreciated that variations from the above described embodiments may still fall within the scope of the invention. For example, the method has been described with reference to people's eyes, for which the reflection from the retina leads to a red region. For some animals, "red-eye" can lead to green or yellow reflections. The method according to the invention may be used to correct for this effect. Indeed, the search for a light, saturated region rather than a region of a particular hue makes the method of the invention particularly suitable for detecting non-red animal "red-eye".

Furthermore, the method has been described for red-eye features in which the highlight region is located exactly in the centre of the red pupil region. However the method will still work for red-eye features whose highlight region is off-centre, or even at the edge of the red region.

Some red-eye features do not have a discrete highlight region, but in these features the whole of the red pupil region has high saturation and lightness values. In such cases the red-eye feature and the highlight region will be the same size, and there may not be any further red part outside the highlight region. In other words, the highlight region 2 and red pupil region 3 will occupy the same area. However, the method described above

17

will still detect such regions as "highlights", with each red region 3 being identified as having the same radius as the highlight. Such features will therefore still be detected using the method according to the invention.

CLAIMS:

1.   A method of processing a digital image, comprising:
        identifying highlight regions of the image having pixels with higher saturation
and/or lightness values than pixels in the regions thereround;
        identifying red-eye features associated with some or all of said highlight regions;
and
        performing red-eye reduction on some or all of the red-eye features.

2.   A method as claimed in claim 1, wherein a single reference pixel in each
highlight region is selected as the central point of an associated red-eye feature, and red-
eye reduction for that red-eye feature is centred on the reference pixel.

3.   A method as claimed in claim 1 or 2, wherein a highlight region is only
identified if there is a sharp change in pixel saturation and/or lightness between the
highlight region and the regions adjacent thereto.

4.   A method as claimed in claim 1, 2 or 3, further comprising eliminating at least
some of the highlight regions as possibilities for red-eye reduction.

5.   A method as claimed in any preceding claim, wherein the red-eye reduction on a
red-eye feature is not carried out if the highlight region associated with that red-eye
feature exceeds a predetermined maximum diameter.

6.   A method as claimed in any preceding claim, further comprising determining
whether each highlight region is substantially linear, and not associating a red-eye
feature with a highlight region if that highlight region is substantially linear.

7.   A method as claimed in any preceding claim, wherein red-eye reduction is not
carried out centred on any red-eye features which overlap each other.

8.   A method as claimed in any preceding claim, further comprising identifying the
hue of pixels in the region surrounding the highlight region for each red-eye feature, and

19

only performing red-eye reduction if the pixels in said region contain more than a predetermined proportion of red.

9. A method as claimed in claim 8, further comprising determining the radius of the red-eye region around each highlight region, the red-eye region having pixels with a hue containing more than said predetermined proportion of red.

10. A method as claimed in claim 9, wherein red-eye reduction is only performed on a red-eye feature if the ratio of radius of the red-eye region to the radius of the highlight region falls within a predetermined range of values.

11. A method as claimed in any preceding claim, wherein the digital image is derived from a photograph, the method further comprising determining whether a flash was fired when the photograph was taken, and not identifying highlight regions or performing red-eye reduction if no flash was fired.

12. A method as claimed in any preceding claim, further comprising determining whether the digital image is monochrome, and not identifying highlight regions or performing red-eye reduction if the digital image is monochrome.

13. A method as claimed in claim 1, 2 or 3, wherein a red-eye feature is associated with each highlight region identified, and red-eye reduction is carried out on all red-eye features.

14. A method of detecting red-eye features in a digital image, comprising:
    identifying highlight regions comprising pixels having higher saturation and/or lightness values than pixels in the regions therearound; and
    determining whether each highlight region corresponds to a red-eye feature on the basis of applying further selection criteria.

15. A method as claimed in claim 14, wherein the further selection criteria include testing the hue of pixels surrounding the highlight region, and determining that the

20

highlight region does not correspond to a red-eye feature if said hue is outside a predetermined range corresponding to red.

16.   A method as claimed in claim 14 or 15, wherein said further selection criteria include identifying the shape of the highlight region, and determining that the highlight region does not correspond to a red-eye feature if said shape is not substantially circular.

17.   A method of reducing the visual effect of red-eye features in a digital image, comprising:

    detecting red-eye features using the method of claim 14, 15 or 16, and

    changing the hue of pixels around each highlight region to reduce the red content of those pixels.

18.   A digital image to which the method of any preceding claim has been applied.

19.   Apparatus arranged to perform the method of any of claims 1 to 17.

20.   A computer storage medium having stored thereon a computer program arranged to perform the method of any of claims 1 to 17.

21.   A method as herein described with reference to the accompanying drawings.
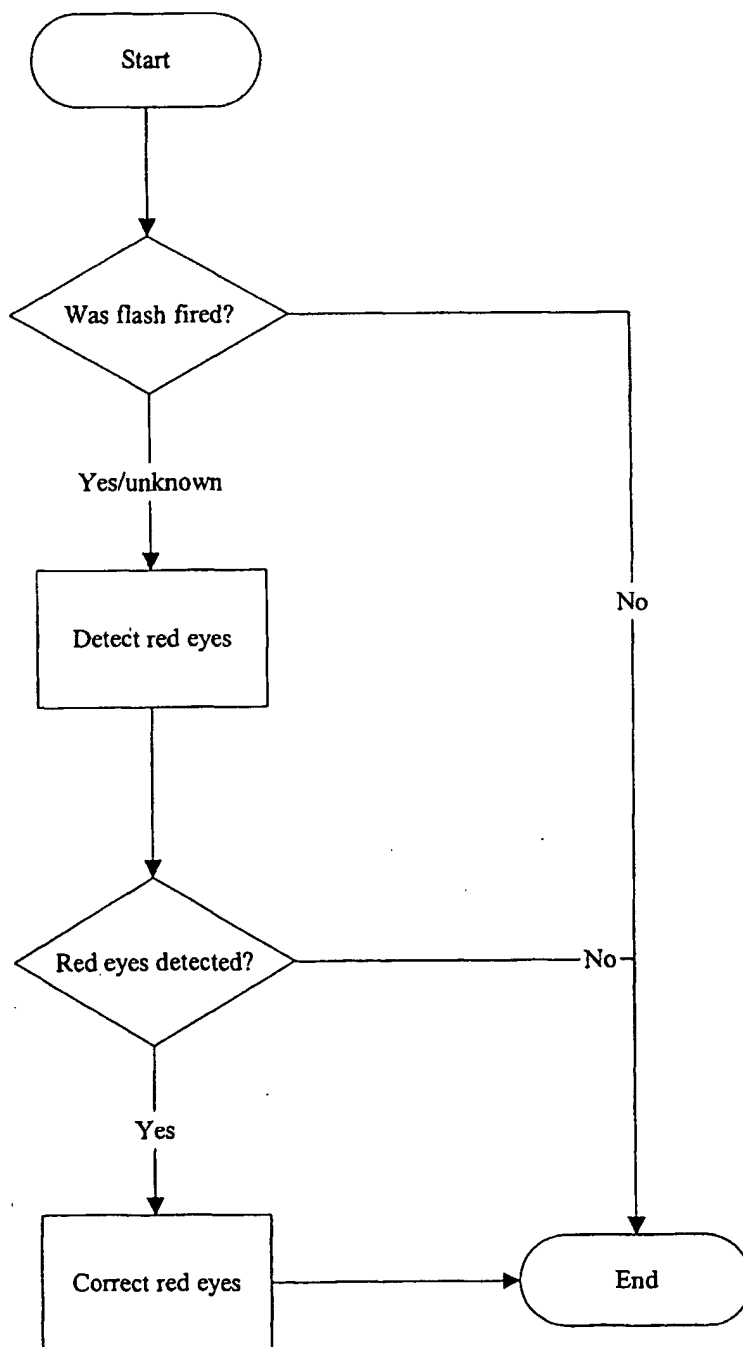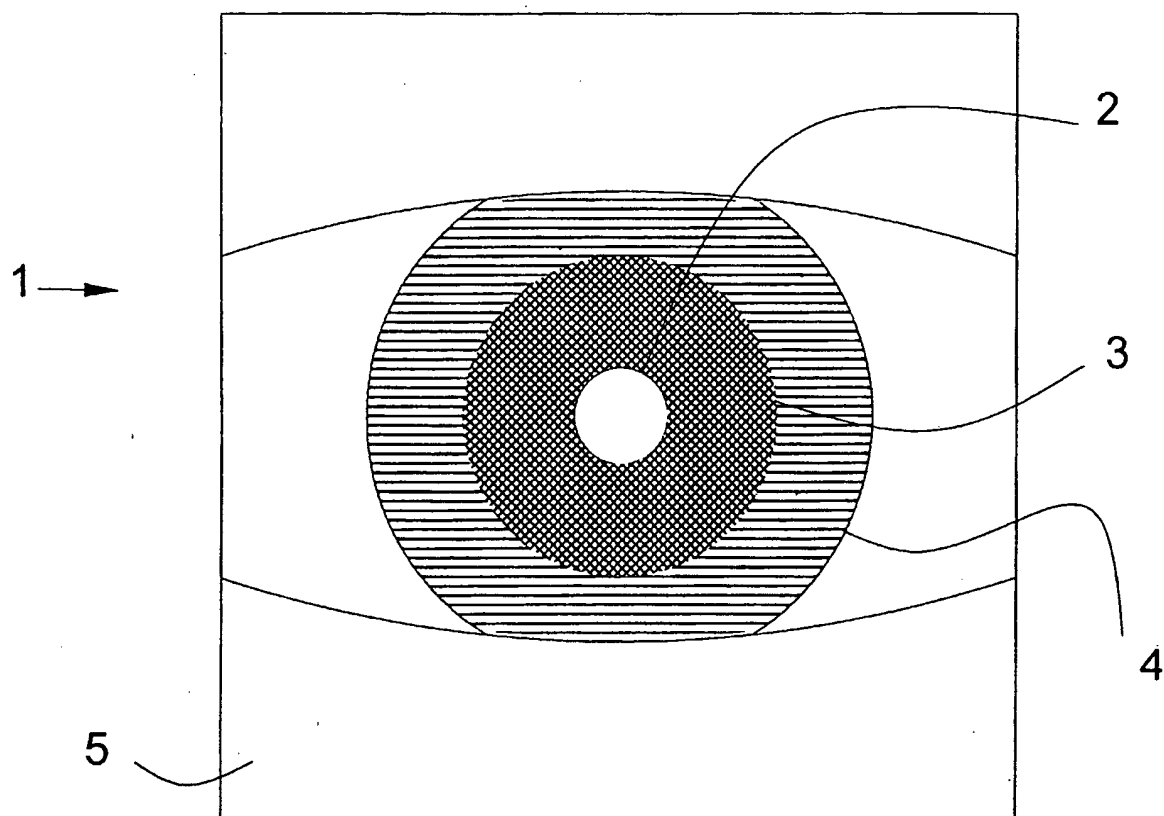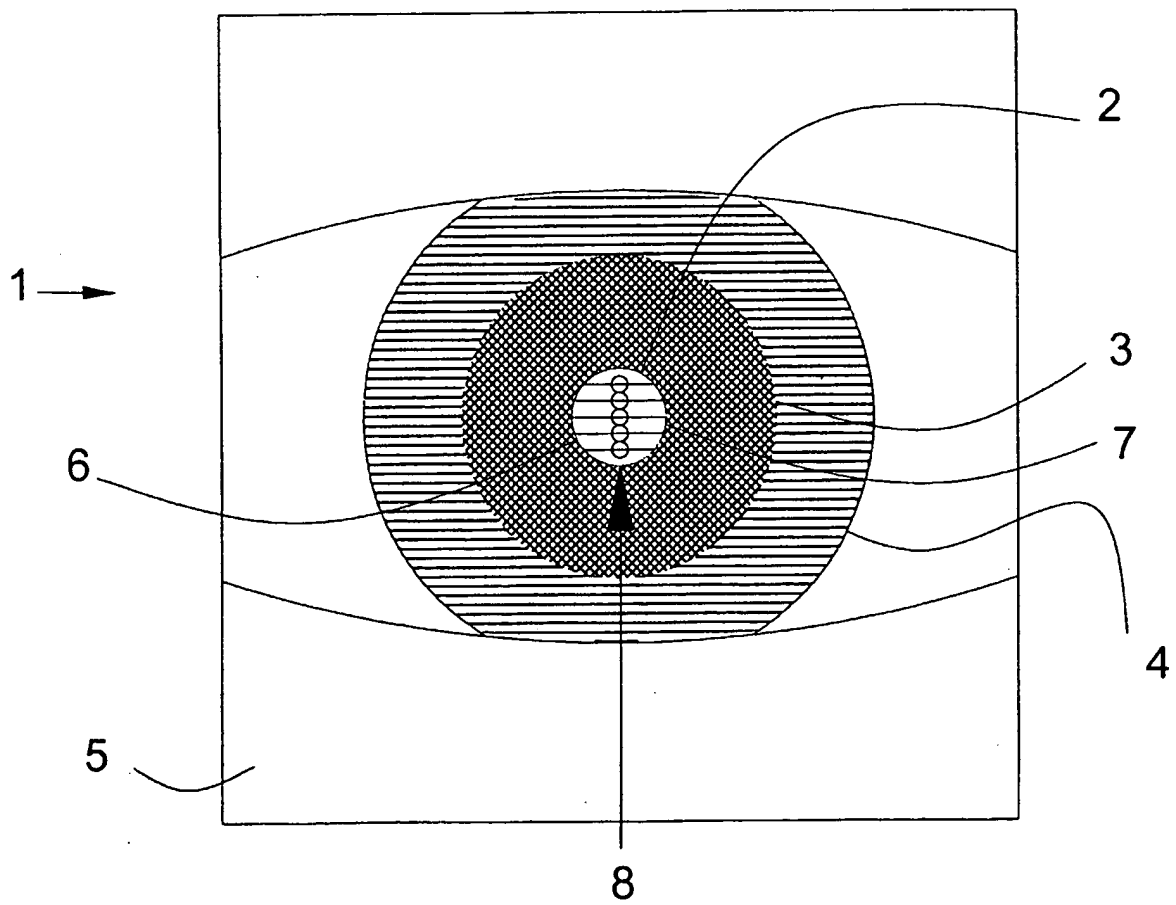
Figure 1

```
                    ┌─────────────┐
                   (    Start     )
                    └──────┬──────┘
                           │
                           ▼
                          ╱ ╲
                        ╱     ╲
                      ╱  Was    ╲────────────────────────┐
                      ╲ flash    ╱                        │
                        ╲ fired? ╱                        │
                          ╲ ╱                             │
                           │                              │
                      Yes/unknown                         │
                           │                              │
                           ▼                             No
                    ┌─────────────┐                       │
                    │ Detect red  │                       │
                    │    eyes     │                       │
                    └──────┬──────┘                       │
                           │                              │
                           ▼                              │
                          ╱ ╲                             │
                        ╱     ╲                           │
                      ╱   Red   ╲──────No────┐            │
                      ╲  eyes     ╱          │            │
                        ╲detected?╱          │            │
                          ╲ ╱                │            │
                           │                 │            │
                          Yes                │            │
                           │                 │            │
                           ▼                 │            ▼
                    ┌─────────────┐          │     ┌────────────┐
                    │ Correct red │──────────┴────▶│    End     │
                    │    eyes     │                └────────────┘
                    └─────────────┘
```

Figure 2

3/5

Figure 3

Figure 4

5/5

Figure 5

```
            ┌──────────┐
            │  Start   │
            └────┬─────┘
                 │
                 ▼
         ┌───────────────┐
         │Detect highlights│
         └───────┬───────┘
                 │
                 ▼
         ┌───────────────┐
         │ Remove linear │
         │  highlights   │
         └───────┬───────┘
                 │
                 ▼
         ┌───────────────┐
         │Remove non-red │
         │or insufficiently│
         │  saturated    │
         │  highlights   │
         └───────┬───────┘
                 │
                 ▼
         ┌───────────────┐
         │Determine radius│
         │  of red-eye   │
         └───────┬───────┘
                 │
                 ▼
         ┌───────────────┐
         │    Remove     │
         │overlapping red-│
         │  eye regions  │
         └───────┬───────┘
                 │
                 ▼
            ┌──────────┐
            │   End    │
            └──────────┘
```
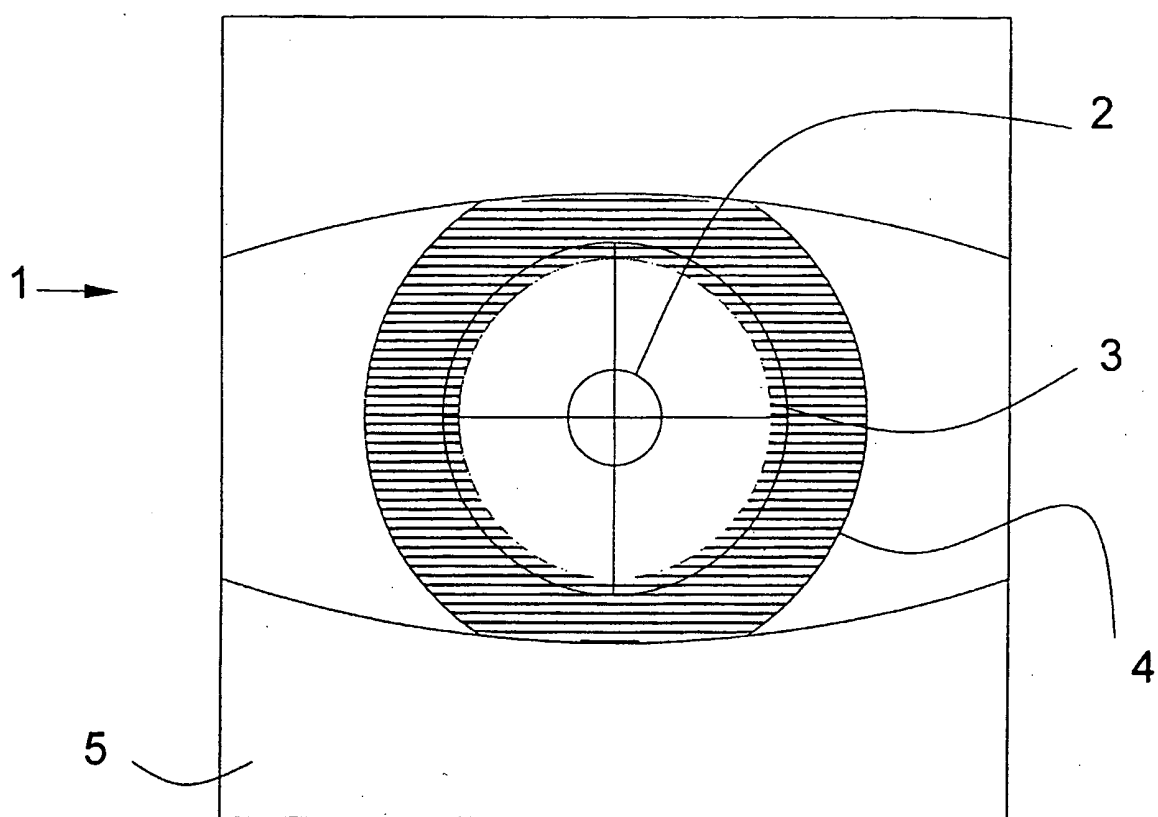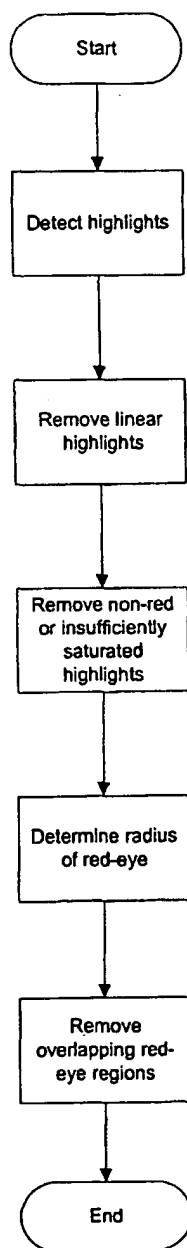
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   H04N1/62   G06T7/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   H04N   G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 911 759 A (HEWLETT PACKARD CO) 28 April 1999 (1999-04-28) abstract paragraphs '0045!,'0049!; claim 1	1-4,6,8, 9,13-21
X A	WO 99 17254 A (POLAROID CORP) 8 April 1999 (1999-04-08) abstract page 4, line 9 - line 19 page 7, line 18 -page 8, line 3	1,3-6,8, 13-21 2,9,10
X A	US 5 748 764 A (BENATI PAUL J ET AL) 5 May 1998 (1998-05-05) column 2, line 3 - line 32 column 5, line 27 - line 37	1-6,8,9, 14-21 7,11-13

-/--

[X] Further documents are listed in the continuation of box C.   [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 October 2002	06/11/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Azaustre Maleno, V

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 884 694 A (ADOBE SYSTEMS INC) 16 December 1998 (1998-12-16)	1,8,9, 14-21
A	page 2, line 26 - line 34 page 5, line 16 - line 21	10
X	PATENT ABSTRACTS OF JAPAN vol. 2000, no. 08, 6 October 2000 (2000-10-06) & JP 2000 134486 A (CANON INC), 12 May 2000 (2000-05-12)	1,3,14
A	abstract	2
A	US 5 130 789 A (DOBBS CHRISTOPHER M ET AL) 14 July 1992 (1992-07-14) abstract	2,6,9,10
A	US 6 252 976 B1 (SCHILDKRAUT JAY S ET AL) 26 June 2001 (2001-06-26) abstract figure 2 column 6, line 36 - line 42	2,5-7,9, 10,16
A	US 6 278 491 B1 (WANG JOHN Y A ET AL) 21 August 2001 (2001-08-21) abstract; figures 4-6	6,7,16
A	US 6 009 209 A (ACKER KRISTIN ET AL) 28 December 1999 (1999-12-28) the whole document	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
EP 0911759	A	28-04-1999	US	6016354 A	18-01-2000
			EP	0911759 A2	28-04-1999
			JP	11232442 A	27-08-1999
WO 9917254	A	08-04-1999	WO	9917254 A1	08-04-1999
US 5748764	A	05-05-1998	US	5432863 A	11-07-1995
			DE	69415886 D1	25-02-1999
			DE	69415886 T2	29-07-1999
			EP	0635972 A2	25-01-1995
			JP	3181472 B2	03-07-2001
			JP	7072537 A	17-03-1995
EP 0884694	A	16-12-1998	US	6204858 B1	20-03-2001
			CA	2236910 A1	30-11-1998
			EP	0884694 A1	16-12-1998
			JP	11073499 A	16-03-1999
JP 2000134486	A	12-05-2000	NONE		
US 5130789	A	14-07-1992	NONE		
US 6252976	B1	26-06-2001	US	6292574 B1	18-09-2001
			EP	0961225 A2	01-12-1999
			JP	2000125320 A	28-04-2000
			EP	0899686 A2	03-03-1999
			JP	11136498 A	21-05-1999
US 6278491	B1	21-08-2001	JP	11284874 A	15-10-1999
US 6009209	A	28-12-1999	NONE		